

Via EFS

Docket No.: 491442003100

(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Marc Timothy JONES et al.

Confirmation No.: 2291

Application No.: 10/817,290

Examiner: Camquy Truong

Filed: April 2, 2004

Art Unit: 2195

For: PREREQUISITE-BASED SCHEDULER

AMENDMENT IN RESPONSE TO FINAL OFFICE ACTION

MS AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

INTRODUCTORY COMMENTS

In response to the Final Office Action dated August 2, 2007, for which a response is due on November 2, 2007, and for which a 1-month extension of time is also requested to extend the time for response from November 2, 2007 to December 3, 2007 (the first business day after December 2, 2007), please amend the above-identified patent application as follows:

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper.

Remarks begin on page 14 of this paper.

AMENDMENTS TO THE CLAIMS

Claim 1 (currently amended): A method for scheduling a plurality of tasks in a processing system having a plurality of defined resources, comprising:

identifying prerequisites for each task, the prerequisites representing all defined resources needed for that task to execute to completion;

identifying one or more higher priority task paths and one or more lower priority task paths, each of the paths having a plurality of tasks in series;

representing each defined resource by one or more resource flags that provide information about that resource;

storing the prerequisites for each task as a collection of resource flags known as a prerequisite row;

storing the system state as a collection of resource flags;

creating a prerequisite table, the prerequisite table having one prerequisite row for each instantiated task arranged in descending order according to in a task priority order determined in accordance with the higher priority task paths and the lower priority task paths, and having one column for each resource flag;

in a descending row order, performing a comparison of the prerequisite row comparing the prerequisites for the task in a given row a particular task against a system state representing a current state of the defined resources until a task is identified for which the comparison reveals that the prerequisites for the identified task are currently available; and

dispatching the identified task.

Claim 2 (previously presented): The method as recited in claim 1, further comprising:

arranging the task priority order such that the tasks in the priority task paths have a higher priority than other tasks.

Claim 3 (original): The method as recited in claim 1, further comprising:
identifying one or more fast paths in a task processing sequence; and
for a given fast path, arranging the task priority order such that the tasks in the given fast path have a higher priority than other tasks.

Claim 4 (original): The method as recited in claim 3, further comprising:
identifying a processing order of the tasks in the given fast path; and
arranging the task priority order such that a last task in the given fast path has a higher priority than a first task in the given fast path.

Claim 5 (original): The method as recited in claim 3, further comprising:
identifying a processing order of the tasks in the given fast path; and
arranging the task priority order such that a first task in the given fast path has a higher priority than a last task in the given fast path.

Claims 6 and 7 (canceled)

Claim 8 (currently amended): The method as recited in claim 1 [[7]], further comprising comparing the prerequisite row for the particular task against the system state by performing a Boolean AND of the prerequisite row and the system state.

Claim 9 (canceled)

Claim 10 (currently amended): The method as recited in claim 1 [[9]], further comprising:
creating a decision tree from the prerequisite table according to the task priority order, the decision tree having one leaf for each instantiated task, one node for each resource, and one edge for each resource flag; and
traversing the decision tree in accordance with the availability of resources.

Claim 11 (original): The method as recited in claim 1, further comprising updating the system state after resources are actually used.

Claim 12 (original): The method as recited in claim 11, wherein after the system state has been updated, the method further comprises implementing fair-share scheduling by resuming the comparison of prerequisite rows against the updated system state at a next row in the prerequisite table.

Claim 13 (original): The method as recited in claim 11, wherein once the system state has been updated, the method further comprises implementing priority scheduling by resuming the comparison of prerequisite rows against the updated system state at a first row in the prerequisite table.

Claim 14 (original): The method as recited in claim 11, further comprising:
grouping consecutive prerequisite rows in the prerequisite table into one or more blocks;
when a task is identified and scheduled for dispatch, identifying the block to which the identified task belongs; and

once the identified task is dispatched and the system state is updated, resuming the comparison at a next row within the identified block, or resuming the comparison at a first row of the prerequisite table if the dispatched task was a last task in the identified block.

Claim 15 (previously presented): The method as recited in claim 1, further comprising:
creating one or more priority flags representing various priority levels as defined priority flag resources in the processing system; and
adding the priority flag resources to the prerequisites of selected tasks in accordance with priority levels desired for those tasks;
wherein the priority flag resources effectively change the task priority order as the prerequisites for the selected tasks are compared against the system state.

Claim 16 (original): The method as recited in claim 1, the defined resources including one or more queues, the method further comprising:

changing a depth of one or more queues to alter the scheduling of tasks having those queues as prerequisites.

Claim 17 (original): The method as recited in claim 1, the defined resources including one or more buffer pools, the method further comprising:

changing a depth of one or more buffer pools to alter the scheduling of tasks having those buffer pools as prerequisites.

Claim 18 (currently amended): The method as recited in claim 1 [[9]], further comprising:
storing the prerequisite rows in M N-bit words, each bit in each N-bit word representing a resource flag and each resource flag being either asserted to indicate that the resource flag is a prerequisite of the task represented by the prerequisite row or deasserted to indicate that the resource flag is not a prerequisite of that task;

storing the system state in M N-bit words, each bit in each N-bit word representing a resource flag, each resource flag being either asserted or deasserted in accordance with the current state of the processing system;

storing an indicator of the last N-bit word containing an asserted resource flag;

comparing the prerequisites and the system state of successive N-bit words only up to the last N-bit word containing an asserted resource flag; and

determining whether the prerequisites for the identified task are currently available using only the compared N-bit words.

Claim 19 (original): The method as recited in claim 18, further comprising:

arranging the resource flags that make up the prerequisite rows and the system state so that when the prerequisite rows are stored in the M-bit words, the number of asserted resource flags appearing in the first M-bit word is maximized.

Claim 20 (currently amended): The method as recited in claim 1 [[9]], further comprising: storing the prerequisite rows in M N-bit words, each bit in each N-bit word representing a resource flag, each resource flag being either asserted to indicate that the resource flag is a prerequisite of the task represented by the prerequisite row or deasserted to indicate that the resource flag is not a prerequisite of that task;

storing the system state in M N-bit words, each bit in each N-bit word representing a resource flag, each resource flag being either asserted or deasserted in accordance with the current state of the processing system;

comparing the prerequisites and the system state of successive N-bit words only as long as the comparison indicates that the prerequisites for the identified task are currently available.

Claim 21 (original): The method as recited in claim 2, further comprising: modifying a scope of the prerequisites for the tasks in a particular priority task path to adjust a processing speed of that priority task path.

Claim 22 (currently amended): In a processing system having a plurality of defined resources, the processing system for executing a plurality of tasks, a computer program for scheduling the plurality of tasks, the computer program being stored on a machine readable medium and executable to perform acts comprising:

identifying prerequisites for each task, the prerequisites representing all defined resources needed for that task to execute to completion;

identifying one or more higher priority task paths and one or more lower priority task paths, each of the paths having a plurality of tasks in series;

representing each defined resource by one or more resource flags that provide information about that resource;

storing the prerequisites for each task as a collection of resource flags known as a prerequisite row;

storing the system state as a collection of resource flags;

creating a prerequisite table, the prerequisite table having one prerequisite row for each instantiated task arranged in descending order according to in a task priority order determined in accordance with the higher priority task paths and the lower priority task paths, and having one column for each resource flag;

in a descending row order, performing a comparison of the prerequisite row comparing the prerequisites for the task in a given row a particular task against a system state representing a current state of the defined resources until a task is identified for which the comparison reveals that the prerequisites for the identified task are currently available; and

scheduling the identified task for dispatch.

Claim 23 (previously presented): The computer program as recited in claim 22, further executable to perform acts comprising:

arranging the task priority order such that the tasks in the priority task paths have a higher priority than other tasks.

Claim 24 (original): The computer program as recited in claim 22, further executable to perform acts comprising:

identifying one or more fast paths in a task processing sequence; and
for a given fast path, arranging the task priority order such that the tasks in the given fast path have a higher priority than other tasks.

Claim 25 (original): The computer program as recited in claim 24, further executable to perform acts comprising:

identifying a processing order of the tasks in the given fast path; and
arranging the task priority order such that a last task in the given fast path has a higher priority than a first task in the given fast path.

Claim 26 (original): The computer program as recited in claim 24, further executable to perform acts comprising:

identifying a processing order of the tasks in the given fast path; and
arranging the task priority order such that a first task in the given fast path has a higher priority than a last task in the given fast path.

Claims 27 and 28 (canceled)

Claim 29 (currently amended): The computer program as recited in claim 22 [[28]], further executable to perform acts comprising comparing the prerequisite row for the particular task against the system state by performing a Boolean AND of the prerequisite row and the system state.

Claim 30 (canceled)

Claim 31 (currently amended): The computer program as recited in claim 22 [[30]], further executable to perform acts comprising:

creating a decision tree from the prerequisite table according to the task priority order, the decision tree having one leaf for each instantiated task, one node for each resource, and one edge for each resource flag; and

traversing the decision tree in accordance with the availability of resources.

Claim 32 (original): The computer program as recited in claim 22, further executable to perform acts comprising updating the system state as resources are actually used.

Claim 33 (original): The computer program as recited in claim 32, wherein after the system state has been updated, the computer program is further executable to perform acts comprising fair-share scheduling by resuming the comparison of prerequisite rows against the updated system state at a next row in the prerequisite table.

Claim 34 (original): The computer program as recited in claim 32, wherein once the system state has been updated, the computer program is further executable to perform acts comprising priority scheduling by resuming the comparison of prerequisite rows against the updated system state at a first row in the prerequisite table.

Claim 35 (original): The computer program as recited in claim 32, further executable to perform acts comprising:

grouping consecutive prerequisite rows in the prerequisite table into one or more blocks; when a task is identified and scheduled for dispatch, identifying the block to which the identified task belongs; and

once the identified task is dispatched and the system state is updated, resuming the comparison at a next row within the identified block, or resuming the comparison at a first row of the prerequisite table if the dispatched task was a last task in the identified block.

Claim 36 (previously presented): The computer program as recited in claim 22, further executable to perform acts comprising:

creating one or more priority flags representing various priority levels as defined priority flag resources in the processing system; and

adding the priority flag resources to the prerequisites of selected tasks in accordance with priority levels desired for those tasks;

wherein the priority flag resources effectively change the task priority order as the prerequisites for the selected tasks are compared against the system state.

Claim 37 (original): The computer program as recited in claim 22, the defined resources including one or more queues, the computer program further executable to perform acts comprising:

changing a depth of one or more queues to alter the scheduling of tasks having those queues as prerequisites.

Claim 38 (original): The computer program as recited in claim 22, the defined resources including one or more buffer pools, the computer program further executable to perform acts comprising:

changing a depth of one or more buffer pools to alter the scheduling of tasks having those buffer pools as prerequisites.

Claim 39 (currently amended): The computer program as recited in claim 22 [[30]], further executable to perform acts comprising:

storing the prerequisite rows in M N-bit words, each bit in each N-bit word representing a resource flag and each resource flag being either asserted to indicate that the resource flag is a prerequisite of the task represented by the prerequisite row or deasserted to indicate that the resource flag is not a prerequisite of that task;

storing the system state in M N-bit words, each bit in each N-bit word representing a resource flag, each resource flag being either asserted or deasserted in accordance with the current state of the processing system;

storing an indicator of the last N-bit word containing an asserted resource flag;

comparing the prerequisites and the system state of successive N-bit words only up to the last N-bit word containing an asserted resource flag; and

determining whether the prerequisites for the identified task are currently available using only the compared N-bit words.

Claim 40 (original): The computer program as recited in claim 39, further executable to perform acts comprising:

arranging the resource flags that make up the prerequisite rows and the system state so that when the prerequisite rows are stored in the M-bit words, the number of asserted resource flags appearing in the first M-bit word is maximized.

Claim 41 (currently amended): The computer program as recited in claim 22 [[30]], further executable to perform acts comprising:

storing the prerequisite rows in M N-bit words, each bit in each N-bit word representing a resource flag, each resource flag being either asserted to indicate that the resource flag is a prerequisite of the task represented by the prerequisite row or deasserted to indicate that the resource flag is not a prerequisite of that task;

storing the system state in M N-bit words, each bit in each N-bit word representing a resource flag, each resource flag being either asserted or deasserted in accordance with the current state of the processing system;

comparing the prerequisites and the system state of successive N-bit words only as long as the comparison indicates that the prerequisites for the identified task are currently available.

Claim 42 (original): The computer program as recited in claim 23, further executable to perform acts comprising:

modifying a scope of the prerequisites for the tasks in a particular priority task path to adjust a processing speed of that priority task path.

Claim 43 (currently amended): A host bus adapter (HBA) comprising the machine readable medium of claim 22
The computer program as recited in claim 22, wherein the machine readable medium on which the computer program is stored is included within a host bus adapter (HBA).

Claim 44 (currently amended): The HBA of claim 43, further comprising
The computer program as recited in claim 43, wherein the HBA further comprises an Internet Small Computer System Interface (iSCSI) or fibre channel (FC) controller circuit.

Claim 45 (currently amended): A host computer comprising the HBA of claim 44
The computer program as recited in claim 44, wherein the HBA is included within a host computer.

Claim 46 (currently amended): A storage area network (SAN) comprising the host computer of claim 45, The computer program as recited in claim 45, wherein the host computer is included within a storage area network (SAN) and wherein an iSCSI or a FC network is coupled to the iSCSI or FC controller circuit, respectively, and one or more storage devices are coupled to the iSCSI or FC network.

REMARKS

Claims 1-46 were pending in the application. Claims 43-46 were rejected under 35 U.S.C. §112, second paragraph, as being indefinite. Claims 1-8, 11-13, 15, 21-29, 32-34, 36 and 42 were rejected under 35 U.S.C. §103(a) as being unpatentable over Flynn et al. in view of Kobayashi and further in view of Miller. Claims 16, 17, 37 and 38 were rejected under 35 U.S.C. §103(a) as being unpatentable over Flynn in view of Kobayashi, and further in view of Miller, as applied as in claim 1, and further in view of Blythe et al. Claims 43-46 were rejected under 35 U.S.C. §103(a) as being unpatentable over Flynn in view of Kobayashi and further in view of Miller, as applied to claim 22, and further in view of Applicant Admitted Prior Art. Claims 9, 10, 14, 18-20, 30, 31, 35 and 39-41 were objected to as being dependent on a rejected base claim, but were indicated to be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Claims 1, 8, 10, 18, 20, 22, 29, 31, 39, 41 and 43-46 have been amended, and claims 6, 7, 9, 27, 28, and 30 have been canceled, leaving claims 1-5, 8, 10-26, 29 and 31-46 presently under consideration. Reexamination and reconsideration of the application in view of the amendments and following remarks are respectfully requested.

Claims 9, 10, 14, 18-20, 30, 31, 35 and 39-41 were objected to as being dependent on a rejected base claim, but were indicated to be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Applicants note with appreciation the indication of allowable subject matter. Claims 1 and 22 have been amended to incorporate the limitations of claims 9 and 30 respectively, and all intervening claims. Therefore, amended claims 1 and 22 are now equivalent to rewriting claims 9 and 10 in independent form, including all of the limitations of the base claim and any intervening claims. Accordingly, Applicants respectfully submit that claims 1 and 22 are now allowable.

In addition, because claims 2-5, 8 and 10-21 depend from claim 1 and claims 23-26, 29 and 31-46 depend from claim 22, Applicants respectfully submit that all presently pending claims are now in condition for allowance.

Claims 43-46 were rejected under 35 U.S.C. §112, second paragraph, as being indefinite. In particular, claims 43-46 were found to be unclear because it was uncertain whether the claims were independent or dependent claims.

Although Applicants maintain that claims 43-46 are proper dependent claims, though unconventional in form, claims 43-46 have nevertheless been amended to a more conventional dependent claim format. With the amendments to claims 43-46, it is respectfully submitted that the rejection of claims 43-46 under 35 U.S.C. §112, second paragraph, has been overcome.

Claims 1-8, 11-13, 15, 21-29, 32-34, 36 and 42 were rejected under 35 U.S.C. §103(a) as being unpatentable over Flynn in view of Kobayashi and further in view of Miller. Claims 6, 7, 27 and 28 have been canceled, rendering their rejections moot. As indicated above, due to the amendments to claims 1 and 22, all presently pending claims are now allowable. Therefore, it is respectfully submitted that the rejection of claims 1-5, 8, 11-13, 15, 21-26, 29, 32-34, 36 and 42 under 35 U.S.C. §103(a) has been overcome.

Claims 16, 17, 37 and 38 were rejected under 35 U.S.C. §103(a) as being unpatentable over Flynn in view of Kobayashi, and further in view of Miller, as applied as in claim 1, and further in view of Blythe. As indicated above, due to the amendments to claims 1 and 22, all presently pending claims are now allowable. Therefore, it is respectfully submitted that the rejection of claims 16, 17, 37 and 38 under 35 U.S.C. §103(a) has been overcome.

Claims 43-46 were rejected under 35 U.S.C. §103(a) as being unpatentable over Flynn in view of Kobayashi and further in view of Miller, as applied to claim 22, and further in view of Applicant Admitted Prior Art. As indicated above, due to the amendments to claims 1 and 22, all presently pending claims are now allowable. Therefore, it is respectfully submitted that the rejection of claims 43-46 under 35 U.S.C. §103(a) has been overcome.

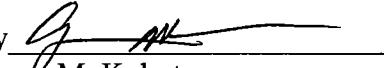
In view of the above, each of the presently pending claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to pass this application to issue.

If, for any reason, the Examiner finds the application other than in condition for allowance, Applicants request that the Examiner contact the undersigned attorney at the Los Angeles telephone number (213) 892-5752 to discuss any steps necessary to place the application in condition for allowance.

In the unlikely event that the transmittal letter is separated from this document and the Patent Office determines that an extension and/or other relief is required, Applicants petition for any required relief including extensions of time and authorizes the Commissioner to charge the cost of such petitions and/or other fees due in connection with the filing of this document to Deposit Account No. 03-1952 referencing Docket No. 491442003100.

Dated: December 3, 2007

Respectfully submitted,

By 
Glenn M. Kubota
Registration No.: 44,197
MORRISON & FOERSTER LLP
555 West Fifth Street, Suite 3500
Los Angeles, California 90013
(213) 892-5200